

AMENDMENT TO THE CLAIMS

1. (CURRENTLY AMENDED) A computer-readable medium having computer-executable instructions for performing ~~steps~~ a method for building a symbol table for storing sort weights for a plurality of linguistic symbols used in a plurality of languages supported by a computer system, the method comprising:

constructing the symbol table to contain a list of code points each uniquely identifying one of the symbols, and a sort weight for the symbol identified by said each code point;

providing a plurality of compression tables, each compression table pertaining to one of the supported languages and having a compression type identifying a number of symbols in a given compression in the compression table and containing compressions of symbols of that compression type, each compression being a grouping of two or more symbols treated as a single sort element for purposes of linguistic sorting such that an order of a given compression in the linguistic sorting is determined based on a compression type of the given compression, a first of the two or more symbols in the given compression and a predefined order of symbols; and

for each code point in the symbol table,

sorting the compression tables using a processor of the computer, based on the sort elements, and identifying a highest compression type for compressions in the compression tables beginning with the symbol identified by ~~said~~ each the code point; and

storing, in the symbol table, a tag for ~~each~~ the code point to indicate said highest compression type for ~~said~~ each the code point, wherein the tag for the code point is stored as a portion of the sort weight of the symbol identified by the code point, and wherein the sort weight of the symbol identified by the

code point comprises a case weight value, and wherein the tag for the code point is stored as part of the case weight value for the code point.

2. (ORIGINAL) A computer-readable medium as in claim 1, wherein the code points are assigned to the symbols according to the Unicode standard.

3-4. (CANCELLED)

5. (ORIGINAL) A computer-readable medium as in claim 1, further comprising computer-executable instructions for performing steps of sorting compressions in each of the compression tables based on combinations of code points of the compressions in said each compression table.

6. (CURRENTLY AMENDED) A method of building a symbol table for storing sort weights for a plurality of linguistic symbols used in a plurality of languages supported by a computer system, comprising:

constructing the symbol table to contain a list of code points each uniquely identifying one of the symbols, and a sort weight for the symbol identified by said each code point;

providing a plurality of compression tables, each compression table pertaining to one of the supported languages and having a compression type and containing compressions of symbols of that compression type, the compression type identifying a number of symbols in a compression, and each compression being a grouping of two or more symbols treated as a single sort element for purposes of linguistic sorting;

for each code point in the symbol table, sorting the compression tables to order the compression and to identify a highest compression type for compressions, the order of the compressions being performed by ordering compressions based on a

first of the two or more symbols and then ordering the compressions based on compression types, beginning with the symbol identified by said each code point; and

storing a tag in the symbol table for each code point to indicate said highest compression type for said each code point, wherein the tag for each code point is stored as a portion of the sort weight of the symbol identified by said each code point, and wherein the sort weight of the symbol identified by said each code point comprises a case weight value, and wherein the tag for said each code point is stored as part of the case weight value for said each code point.

7. (ORIGINAL) A method as in claim 6, wherein the code points are assigned to the symbols according to the Unicode standard.

8-9. (CANCELLED)

10. (CURRENTLY AMENDED) A method as in claim 6, further ~~including the step of comprising~~ sorting compressions in each of the compression tables based on combinations of code points of the compressions in said each compression table.

11. (PREVIOUSLY PRESENTED) A computer-readable medium having computer-executable instructions for performing steps for a computer search program to carry out a linguistic sorting operation, comprising:

receiving an input string containing a plurality of letters used in a given language;

for a first letter in a combination of letters in the input string, referencing a symbol table to obtain a highest compression type for compressions beginning with said first letter, each compression being a grouping of two or more letter treated as a single sort element for purposes of linguistic sorting and the compression type

identifying a number of letters in a given compression, the symbol table having a list of code points each uniquely identifying a letter and a sort weight for the letter identified by said each code point;

performing a binary search through each of a plurality of compression tables containing compressions for the given language to find a matching compression that matches said combination of letters in the input string, wherein the plurality of compression tables are searched in a descending order of compression types of the compression tables starting with a compression table having a compression type equal to said highest compression type for said first letter.

12. (CURRENTLY AMENDED) A computer-readable medium as in claim ~~11~~25, wherein the compressions in each of the compression tables are sorted according to code points for letters forming the compressions.

13. (PREVIOUSLY PRESENTED) A computer-readable medium as in claim 12, wherein each code point in the symbol table includes a tag indicating a highest compression type for said each code point, and wherein said step of referencing retrieves the tag for the code point identifying said first letter.

14. (ORIGINAL) A computer-readable medium as in claim 13, wherein the tag for each code point in the symbol table is stored as a portion of the sort weight for said each code point.

15. (CURRENTLY AMENDED) A computer-readable medium as in claim ~~14~~25, wherein the code points in the symbol table are assigned to letters according to a Unicode standard.

16. (CURRENTLY AMENDED) A computer-readable medium as in claim ~~11~~25, wherein the computer-executable instructions for performing a binary search form a module that is called for searching each of the compression tables.

17. (CURRENTLY AMENDED) A computer-readable medium as in claim ~~11~~25, having further computer-executable instructions for storing a sort weight for the matching compression.

18-24. CANCELLED.

25. (NEW) The computer-readable medium of claim 11, wherein performing a binary search comprises performing a binary search using a processor of the computer.

26. (NEW) The computer-readable medium of claim 11, wherein the combination of letters comprises a first combination of letters in the input string, the method further comprising:

identifying a next combination of letters in the input string, the next combination of letters following the first combination of letters in the input string;

for a first letter in the next combination of letters in the input string, referencing the symbol table to obtain a highest compression type for compressions beginning with said first letter in the next combination of letters;

performing a binary search through each of the plurality of compression tables containing compressions for the given language to find a matching compression that matches said next combination of letters in the input string, wherein the plurality of compression tables are searched in a descending order of compression types of the compression tables starting with a compression table having a compression type equal to said highest compression type for said first letter in the next combination of letters in the input string.